179. (New) A method for programming a system having a cellular structure, comprising:

extracting a control flow graph of a program;

separating the control flow graph into a plurality of subgraphs; and

distributing the plurality of subgraphs among a plurality of programmable hardware modules.

180. (New) A method for programming a system having a cellular structure, comprising:

extracting a data flow graph of a program;

separating the data flow graph into a plurality of subgraphs; and

distributing the plurality of subgraphs among a plurality of hardware modules.

181. (New) A method for programming a system having a cellular structure, comprising:

extracting from a program at least one of a data flow graph and a control flow graph;

separating the at least one of the graphs into a plurality of subgraphs; and

distributing the plurality of subgraphs among a plurality of hardware modules.

182. (New) The method according to claim 181, wherein the separating step including separating the at least one of the graphs so as minimize connections between the plurality of subgraphs.

183. (New) The method of claim 181, wherein the separating step includes separating the at least one the graphs into

the plurality of subgraphs so that data transmission between the plurality of subgraphs is minimized.

184. (New) The method of claim 181, wherein the separating step includes separating the at least one of the graphs into the plurality of subgraphs so that no loop-back is obtained between the plurality of subgraphs.

185. (New) The method of claim 181, wherein the separating step includes separating the at least one of the graphs into the plurality of subgraphs so that the subgraphs match resources of the hardware modules.

186. (New) The method of claim 181, further comprising:

inserting memory elements between the plurality of subgraphs, the memory elements adapted to save data passed between subgraphs.

187. (New) The method of claim 181, wherein each of the plurality of subgraphs includes nodes, the method further comprising:

transmitting status signals between nodes within one of the subgraphs so that a state of each individual one of the nodes of the one of the subgraphs is available to each of the other nodes of the one of the subgraphs.

188. (New) The method of claim 181, wherein each of the plurality of subgraphs includes nodes, the method further comprising:

transmitting status signals from a first node of at least one of the plurality of subgraphs to a higher-level unit adapted to control configuration of the plurality of hardware modules so as to trigger reconfiguration.

189. (New) The method of claim 181, wherein the extracting step includes, for a conditional instruction, extracting a plurality of different subgraphs, each representing a different instruction path, one of the different subgraphs being executed depending on an evaluation of the conditional instruction.

190. (New) A method of executing a program on a system having a cellular structure, comprising:

transmitting a data signal from a first cell to a second cell; and

transmitting a status with the data signal, the status indicating whether the data signal is valid.

191. (New) The method of claim 190, further comprising:

receiving a valid data signal at the second cell; and

acknowledging receipt of the valid data signal.

192. (New) The method of claim 191, further comprising,

transmitting by the second cell an indication that a signal is expected.

193. (New) The method of claim 192, further comprising:

transmitting by the first cell an indication that the first cell is transmitting the expected signal.

194. (New) A method of executing a program on a system having a cellular structure, the cell structure including a plurality of cells, the method comprising:

forming a plurality of subgraphs based on a program;

computing a first part of a first one of the subgraphs with a first cell;

after the computing, reconfiguring the first cell for computation of a first part of a second one of the subgraphs; and

simultaneously with the reconfiguring, computing a second part of the first subgraph with a second cell.

195. (New) The method of claim 194, further comprising:

storing configurations for the first one of the subgraphs and the second one of the subgraphs in configuration registers associated with the first cell.

196. (New) The method of claim 195, further comprising:

marking unconfigured ones of the configuration registers as unconfigured.

197. (New) The method of claim 194, further comprising:

selecting a configuration for the first cell based on a status signal generated by the cell structure.

198. (New) The method of claim 194, further comprising:

selecting a configuration for the first cell based on a status signal generated by a higher-level loading unit.

199. (New) The method of claim 194, further comprising:

selecting a configuration for the first cell based on an externally generated status signal.

200. (New) The method of claim 194, further comprising:

selecting a configuration for the first cell as a function of a present configuration of the first cell and a